Logit, Probit, and Multinomial Logit models in R
(v. 3.5)

Oscar Torres-Reyna
otorres@princeton.edu

December 2014

http://www.princeton.edu/~otorres/
If outcome or dependent variable is binary and in the form 0/1, then use logit or probit models. Some examples are:

<table>
<thead>
<tr>
<th>Question</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Did you vote in the last election?</td>
<td>0 ‘No’ 1 ‘Yes’</td>
</tr>
<tr>
<td>Do you prefer to use public transportation or to drive a car?</td>
<td>0 ‘Prefer to drive’ 1 ‘Prefer public transport’</td>
</tr>
</tbody>
</table>

If outcome or dependent variable is categorical but are ordered (i.e. low to high), then use ordered logit or ordered probit models. Some examples are:

<table>
<thead>
<tr>
<th>Question</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you agree or disagree with the President?</td>
<td>1 ‘Disagree’ 2 ‘Neutral’ 3 ‘Agree’</td>
</tr>
<tr>
<td>What is your socioeconomic status?</td>
<td>1 ‘Low’ 2 ‘Middle’ 3 ‘High’</td>
</tr>
</tbody>
</table>

If outcome or dependent variable is categorical without any particular order, then use multinomial logit. Some examples are:

<table>
<thead>
<tr>
<th>Question</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>If elections were held today, for which party would you vote?</td>
<td>1 ‘Democrats’ 2 ‘Independent’ 3 ‘Republicans’</td>
</tr>
<tr>
<td>What do you like to do on the weekends?</td>
<td>1 ‘Rest’ 2 ‘Go to movies’ 3 ‘Exercise’</td>
</tr>
</tbody>
</table>
Logit model

# Getting sample data
library(foreign)
mydata <- read.dta("https://dss.princeton.edu/training/Panel101.dta")

# Running a logit model
logit <- glm(y_bin ~ x1 + x2 + x3, family=binomial(link="logit"), data=mydata)

summary(logit)

Call:
glm(formula = y_bin ~ x1 + x2 + x3, family = binomial(link = "logit"),
data = mydata)

Deviance Residuals:
     Min       1Q   Median       3Q      Max
-2.0277   0.2347   0.5542   0.7016   1.0839

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.4262     0.6390   0.667   0.5048
x1           0.8618     0.7840   1.099   0.2717
x2           0.3665     0.3082   1.189   0.2343
x3           0.7512     0.4548   1.652   0.0986 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 70.056  on 69  degrees of freedom
Residual deviance: 65.512  on 66  degrees of freedom
AIC: 73.512

Number of Fisher Scoring iterations: 5

The \textbf{Pr(>|z|)} column shows the two-tailed p-values testing the null hypothesis that the coefficient is equal to zero (i.e. no significant effect). The usual value is 0.05, by this measure none of the coefficients have a significant effect on the log-odds ratio of the dependent variable. The coefficient for \textit{x3} is significant at 10% (<0.10).

The \textbf{z value} also tests the null that the coefficient is equal to zero. For a 5\% significance, the \textit{z-value} should fall outside the ±1.96.

The \textbf{Estimate} column shows the coefficients in log-odds form. When \textit{x3} increase by one unit, the expected change in the log odds is 0.7512. What you get from this column is whether the effect of the predictors is positive or negative. See next page for an extended explanation.
# The stargazer() function from the package stargazer allows a publication quality of the logit model.

# The model will be saved in the working directory under the name `logit.htm` which you can open with Word or any other word processor.

```r
library(stargazer)
stargazer(logit, type="html", out="logit.htm")
```

<table>
<thead>
<tr>
<th>Dependent variable: y_bin</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
</tr>
<tr>
<td>x2</td>
</tr>
<tr>
<td>x3</td>
</tr>
<tr>
<td>Constant</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Observations</td>
<td>70</td>
</tr>
<tr>
<td>Log Likelihood</td>
<td>-32.756</td>
</tr>
<tr>
<td>Akaike Inf. Crit.</td>
<td>73.512</td>
</tr>
</tbody>
</table>

**Note:** Use the option `type = "text"` if you want to see the results directly in the RStudio console.
# Estimating the odds ratio by hand

```r
cbind(Estimate=round(coef(logit),4),
      OR=round(exp(coef(logit)),4))
```

<table>
<thead>
<tr>
<th>Estimate</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.4262 1.5314</td>
</tr>
<tr>
<td>x1</td>
<td>0.8618 2.3674</td>
</tr>
<tr>
<td>x2</td>
<td>0.3665 1.4427</td>
</tr>
<tr>
<td>x3</td>
<td>0.7512 2.1196</td>
</tr>
</tbody>
</table>

The `Estimate` column shows the coefficients in log-odds form. When `x3` increases by one unit, the expected change in the log odds is 0.7512. Let's hold `x1` and `x2` constant at their means, and vary `x3` with values 1, 2, and 3, to get the predicted log-odds given each of the three values of `x3`:

```r
r1 <- logit$coeff[1] + logit$coeff[2]*mean(mydata$x1) +
      logit$coeff[3]*mean(mydata$x2) +
      logit$coeff[4]*1
> r1
1.784902

r2 <- logit$coeff[1] + logit$coeff[2]*mean(mydata$x1) +
      logit$coeff[3]*mean(mydata$x2) +
      logit$coeff[4]*2
> r2
2.536113

r3 <- logit$coeff[1] + logit$coeff[2]*mean(mydata$x1) +
      logit$coeff[3]*mean(mydata$x2) +
      logit$coeff[4]*3
> r3
3.287325
```

**When `x3` increases from 1 to 2, the log-odds increases:**
```
r2-r1
0.7512115
```

**When `x3` increases from 2 to 3, the log-odds increases:**
```
r3-r2
0.7512115
```

Which corresponds to the estimate for `x3` above.

The odds ratio, is the exponentiation of the difference of the log-odds
```
> exp(r2-r1)
2.119566
```

Or, the ratio of the exponentiation of each of the log-odds.
```
> exp(r2)/exp(r1)
2.119566
```

Which corresponds to the OR value for `x3` above.
# Relative risk ratios allow an easier interpretation of the logit coefficients. They are the exponentiated value of the logit coefficients.

\[
\text{logit.or} = \exp(\text{coef(\text{logit}))}
\]

\[
\begin{array}{lcccc}
\text{(Intercept)} & x1 & x2 & x3 \\
1.531417 & 2.367352 & 1.442727 & 2.119566 \\
\end{array}
\]

library(stargazer)

stargazer(logit, type="html", coef=list(logit.or), p.auto=FALSE, out="logitor.htm")

<table>
<thead>
<tr>
<th>Dependent variable:</th>
</tr>
</thead>
<tbody>
<tr>
<td>y_bin</td>
</tr>
</tbody>
</table>

Keeping all other variables constant, when x1 increases one unit, it is 2.367 times more likely to be in the 1 category. In other words, the odds of being in the 1 category (as opposed to the 0 category) are 136% higher when x1 move one unit (2.36 – 1). The coefficient, however, is not significant.

NOTE: Use the option `type = "text"` if you want to see the results directly in the RStudio console.
Logit model: predicted probabilities

The logit model can be written as (Gelman and Hill, 2007):

\[
\Pr(y_i = 1) = \text{Logit}^{-1}(X_i\beta)
\]

In the example:

```r
logit <- glm(y_bin ~ x1 + x2 + x3, family=binomial(link="logit"), data=mydata)

coef(logit)

(Intercept)       x1       x2       x3
 0.4261935  0.8617722  0.3665348  0.7512115

Pr(y_i = 1) = \text{Logit}^{-1}(0.4261935 + 0.8617722*x1 + 0.3665348*x2 + 0.7512115*x3)
```

Estimating the probability at the mean point of each predictor can be done by inverting the logit model. Gelman and Hill provide a function for this (p. 81), also available in the R package `arm`:

```r
invlogit = function (x) {1/(1+exp(-x))}
invlogit(coef(logit)[1] +
    coef(logit)[2]*mean(mydata$x1)+
    coef(logit)[3]*mean(mydata$x2)+
    coef(logit)[4]*mean(mydata$x3))

Pr(y_i = 1) = 0.8328555
```
Logit model: predicted probabilities

Adding categorical variable, the model would be:

```r
logit.cat <- glm(y_bin ~ x1 + x2 + x3 + opinion,
                 family=binomial(link="logit"),
                 data=mydata)
```

```
coef(logit.cat)
             (Intercept)            x1            x2            x3 opinionAgree opinionDisag opinionStr disag
0.8816118     1.1335562     0.3021217     0.3976276   -1.9163569    0.3270627    0.6891686
```

Estimating the probability when opinion = ‘Agree’

```r
invlogit = function (x) {1/(1+exp(-x))}

invlogit(coef(logit.cat)[1] +
         coef(logit.cat)[2]*mean(mydata$x1) +
         coef(logit.cat)[3]*mean(mydata$x2) +
         coef(logit.cat)[4]*mean(mydata$x3) +
         coef(logit.cat)[5]*1)

Pr(y_i = 1 | opinion= “Agree”) = 0.5107928
```
Logit model: predicted probabilities

$$invlogit = function \ (x) \ \{1/(1+\exp(-x))\}$$

Estimating the probability when opinion = ‘Disagree’

$$invlogit(coef(logit.cat)[1]+coef(logit.cat)[2]*mean(mydata$x1)+coef(logit.cat)[3]*mean(mydata$x2)+coef(logit.cat)[4]*mean(mydata$x3)+coef(logit.cat)[6]*1)$$

$$Pr(y_i = 1| \ \text{opinion} = “Disagree” ) = 0.9077609$$

Estimating the probability when opinion = ‘Strongly disagree’

$$invlogit(coef(logit.cat)[1]+coef(logit.cat)[2]*mean(mydata$x1)+coef(logit.cat)[3]*mean(mydata$x2)+coef(logit.cat)[4]*mean(mydata$x3)+coef(logit.cat)[7]*1)$$

$$Pr(y_i = 1| \ \text{opinion} = “Strongly disagree” ) = 0.933931$$

Estimating the probability when opinion = ‘Strongly agree’

$$invlogit(coef(logit.cat)[1]+coef(logit.cat)[2]*mean(mydata$x1)+coef(logit.cat)[3]*mean(mydata$x2)+coef(logit.cat)[4]*mean(mydata$x3))$$

$$Pr(y_i = 1| \ \text{opinion} = “Strongly agree” ) = 0.8764826$$
Logit model: predicted probabilities

Another way to estimate the predicted probabilities is by setting initial conditions.

Getting predicted probabilities holding all predictors or independent variables to their means.

```r
allmean <- data.frame(x1=mean(mydata$x1),
                      x2=mean(mydata$x2),
                      x3=mean(mydata$x3))

allmean
   x1    x2    x3
1 0.648 0.134 0.762
```

After estimating the logit model and creating the dataset with the mean values of the predictors, you can use the `predict()` function to estimate the predicted probabilities (for help/details type `?predict.glm`), and add them to the `allmean` dataset.

```r
allmean$pred.prob <- predict(logit, newdata=allmean, type="response")

allmean
   x1    x2    x3  pred.prob
1 0.648 0.134 0.762 0.833
```

When all predictor values are held to their means, the probability of $y = 1$ is 83%. 

Creating a new dataset with the mean values of the predictors

The object with the logit coefficients

Dataset with the conditions

Requesting predicted probabilities
Logit model: predicted probabilities with categorical variable

logit <- glm(y_bin ~ x1+x2+x3+opinion, family=binomial(link="logit"), data=mydata)

To estimate the predicted probabilities, we need to set the initial conditions. Getting predicted probabilities holding all predictors or independent variables to their means for each category of categorical variable ‘opinion’:

allmean <- data.frame(x1=rep(mean(mydata$x1),4),
                      x2=rep(mean(mydata$x2),4),
                      x3=rep(mean(mydata$x3),4),
                      opinion=as.factor(c("Str agree","Agree","Disag","Str disag")))

allmean

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>opinion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Str agree</td>
</tr>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Agree</td>
</tr>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Disag</td>
</tr>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Str disag</td>
</tr>
</tbody>
</table>

allmean <- cbind(allmean,predict(logit, newdata=allmean, type="response", se.fit=TRUE))

allmean

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>opinion</th>
<th>fit</th>
<th>se.fit</th>
<th>residual.scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Str agree</td>
<td>0.8764826</td>
<td>0.07394431</td>
<td>1</td>
</tr>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Agree</td>
<td>0.5107928</td>
<td>0.15099064</td>
<td>1</td>
</tr>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Disag</td>
<td>0.9077609</td>
<td>0.06734568</td>
<td>1</td>
</tr>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Str disag</td>
<td>0.9339310</td>
<td>0.06446677</td>
<td>1</td>
</tr>
</tbody>
</table>

(continue next page)
Logit model: predicted probabilities with categorical variable

# Renaming "fit" and "se.fit" columns
names(allmean)[names(allmean)=="fit"] = "prob"

names(allmean)[names(allmean)=="se.fit"] = "se.prob"

# Estimating confidence intervals
allmean$ll = allmean$prob - 1.96*allmean$se.prob

allmean$ul = allmean$prob + 1.96*allmean$se.prob

allmean

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>opinion</th>
<th>prob</th>
<th>se.prob</th>
<th>residual.scale</th>
<th>ll</th>
<th>ul</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Str agree</td>
<td>0.8764826</td>
<td>0.07394431</td>
<td>1</td>
<td>0.7315518</td>
<td>1.0214134</td>
</tr>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Agree</td>
<td>0.5107928</td>
<td>0.15099064</td>
<td>1</td>
<td>0.2148511</td>
<td>0.8067344</td>
</tr>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Disag</td>
<td>0.9077609</td>
<td>0.06734568</td>
<td>1</td>
<td>0.7757634</td>
<td>1.0397585</td>
</tr>
<tr>
<td>0.6480006</td>
<td>0.1338694</td>
<td>0.761851</td>
<td>Str disag</td>
<td>0.9339310</td>
<td>0.06446677</td>
<td>1</td>
<td>0.8075762</td>
<td>1.0602859</td>
</tr>
</tbody>
</table>

(continue next page)
Logit model: predicted probabilities with categorical variable

# Plotting predicted probabilities and confidence intervals using ggplot2

library(ggplot2)

ggplot(allmean, aes(x=opinion, y = prob)) +
  geom_errorbar(aes(ymin = ll, ymax = ul), width = 0.2, lty=1, lwd=1, col="red") +
  geom_point(shape=18, size=5, fill="black") +
  scale_x_discrete(limits = c("Str agree","Agree","Disag","Str disag")) +
  labs(title= " Predicted probabilities", x="Opinion", y="Pr(y=1)", caption = "add footnote here") +
  theme(plot.title = element_text(family = "sans", face="bold", size=13, hjust=0.5),
        axis.title = element_text(family = "sans", size=9),
        plot.caption = element_text(family = "sans", size=5))
# Using package -mfx-
# See http://cran.r-project.org/web/packages/mfx/mfx.pdf
install.packages("mfx")  #Do this only once
library(mfx)
logitmfx(y_bin ~ x1+x2+x3, data=mydata)

Call:
logitmfx(formula = y_bin ~ x1 + x2 + x3, data = mydata)

Marginal Effects:

|     | dF/dx | Std. Err. |     z  | P>|z| |
|-----|-------|-----------|-------|-----|
| x1  | 0.119965 | 0.104836  | 1.1443 | 0.25249 |
| x2  | 0.051024  | 0.041155  | 1.2398 | 0.21504 |
| x3  | 0.104574  | 0.053890  | 1.9405 | 0.05232 . |

Marginal effects show the change in probability when the predictor or independent variable increases by one unit. For continuous variables this represents the instantaneous change given that the ‘unit’ may be very small. For binary variables, the change is from 0 to 1, so one ‘unit’ as it is usually thought.
# Getting sample data
library(foreign)
mydata <- read.dta("https://dss.princeton.edu/training/Panel101.dta")

# Loading library -MASS-
library(MASS)

# Running the ordered logit model
m1 <- polr(opinion ~ x1 + x2 + x3, data=mydata, Hess=TRUE)

summary(m1)
Call:
polr(formula = opinion ~ x1 + x2 + x3, data = mydata, Hess = TRUE)

Coefficients:
       Value Std. Error t value
x1 0.98140     0.5641  1.7397
x2 0.24936     0.2086  1.1954
x3 0.09089     0.1549  0.5867

Intercepts:
       Value Std. Error t value
Str agree|Agree -0.2054  0.4682 -0.4388
Agree|Disag  0.7370  0.4697  1.5690
Disag|Str disag 1.9951  0.5204  3.8335

Residual Deviance: 189.6382
AIC: 201.6382
Ordinal logit model: p-values

# Getting coefficients and p-values

m1.coef <- data.frame(coef(summary(m1)))

m1.coef$pval = round((pnorm(abs(m1.coef$t.value), lower.tail = FALSE) * 2),2)

m1.coef

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Std..Error</th>
<th>t.value</th>
<th>pval</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0.9814</td>
<td>0.5641</td>
<td>1.7397</td>
<td>0.08</td>
</tr>
<tr>
<td>x2</td>
<td>0.2494</td>
<td>0.2086</td>
<td>1.1954</td>
<td>0.23</td>
</tr>
<tr>
<td>x3</td>
<td>0.0909</td>
<td>0.1549</td>
<td>0.5867</td>
<td>0.56</td>
</tr>
<tr>
<td>Str agree</td>
<td>Agree</td>
<td>-0.2054</td>
<td>0.4682</td>
<td>-0.4388</td>
</tr>
<tr>
<td>Agree</td>
<td>Disag</td>
<td>0.7369</td>
<td>0.4697</td>
<td>1.5690</td>
</tr>
<tr>
<td>Disag</td>
<td>Str disag</td>
<td>1.9951</td>
<td>0.5204</td>
<td>3.8335</td>
</tr>
</tbody>
</table>
# The stargazer() function from the package -stargazer allows a publication quality of the logit model.
# The model will be saved in the working directory under the name ‘ml.htm’ which you can open with Word or any other word processor.

```
library(stargazer)
stargazer(m1, type="html", out="ml.htm")
```

<table>
<thead>
<tr>
<th></th>
<th>Dependent variable:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>opinion</td>
</tr>
<tr>
<td>x1</td>
<td>0.981*</td>
</tr>
<tr>
<td></td>
<td>(0.564)</td>
</tr>
<tr>
<td>x2</td>
<td>0.249</td>
</tr>
<tr>
<td></td>
<td>(0.209)</td>
</tr>
<tr>
<td>x3</td>
<td>0.091</td>
</tr>
<tr>
<td></td>
<td>(0.155)</td>
</tr>
<tr>
<td>Observations</td>
<td>70</td>
</tr>
<tr>
<td>Note:</td>
<td>*p **p ***p&lt;0.01</td>
</tr>
</tbody>
</table>

**NOTE:** Use the option type = "text" if you want to see the results directly in the RStudio console.
# Relative risk ratios allow an easier interpretation of the logit coefficients. They are the exponentiated value of the logit coefficients.

m1.or=exp(coef(m1))
m1.or

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.668179</td>
<td>1.283198</td>
<td>1.095150</td>
</tr>
</tbody>
</table>

library(stargazer)
stargazer(m1, type="html", coef=list(m1.or), p.auto=FALSE, out="m1or.htm")

<table>
<thead>
<tr>
<th>Dependent variable:</th>
<th>opinion</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>2.668*</td>
</tr>
<tr>
<td></td>
<td>(0.564)</td>
</tr>
<tr>
<td>x2</td>
<td>1.283</td>
</tr>
<tr>
<td></td>
<td>(0.209)</td>
</tr>
<tr>
<td>x3</td>
<td>1.095</td>
</tr>
<tr>
<td></td>
<td>(0.155)</td>
</tr>
<tr>
<td>Observations</td>
<td>70</td>
</tr>
</tbody>
</table>

Keeping all other variables constant, when x1 increases one unit, it is 2.668 times more likely to be in a higher category. In other words, the odds of moving to a higher category in the outcome variable is 166% when x1 move one unit (2.66 – 1). The coefficient is significant.

**NOTE:** Use the option type = "text" if you want to see the results directly in the RStudio console.
Ordinal logit model: predicted probabilities

# Use "probs" for predicted probabilities

m1.pred <- predict(m1, type="probs")
summary(m1.pred)

<table>
<thead>
<tr>
<th></th>
<th>Str agree</th>
<th>Agree</th>
<th>Disag</th>
<th>Str disag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>0.1040</td>
<td>0.1255</td>
<td>0.1458</td>
<td>0.07418</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>0.2307</td>
<td>0.2038</td>
<td>0.2511</td>
<td>0.17350</td>
</tr>
<tr>
<td>Median</td>
<td>0.2628</td>
<td>0.2144</td>
<td>0.2851</td>
<td>0.23705</td>
</tr>
<tr>
<td><strong>Mean</strong></td>
<td><strong>0.2869</strong></td>
<td><strong>0.2124</strong></td>
<td><strong>0.2715</strong></td>
<td><strong>0.22923</strong></td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.3458</td>
<td>0.2271</td>
<td>0.2949</td>
<td>0.26968</td>
</tr>
<tr>
<td>Max.</td>
<td>0.5802</td>
<td>0.2313</td>
<td>0.3045</td>
<td>0.48832</td>
</tr>
</tbody>
</table>

The bold numbers are the predicted probabilities of each category when all predictors are at their mean value
Ordinal logit model: predicted probabilities

# At specific values, example x1 and x2 at their means, and x3 = 1 and x3 = 2.
# Use "probs" for predicted probabilities given specific predictors

```r
setup1 <- data.frame(x1=rep(mean(mydata$x1),2),
                     x2=rep(mean(mydata$x2),2),
                     x3=c(1,2))
```

```r
setup1
   x1        x2 x3
1 0.6480006 0.1338694  1
2 0.6480006 0.1338694  2
```

```r
setup1[, c("pred.prob")]<- predict(m1, newdata=setup1, type="probs")
setup1
   x1        x2 x3  pred.prob.Str agree pred.prob.Agree pred.prob.Disag pred.prob.Str disag
1 0.6480006 0.1338694  1 0.2757495       0.2184382       0.2804806       0.2253318
2 0.6480006 0.1338694  2 0.2579719       0.2135235       0.2869123       0.2415923
```

# Use "class" for the predicted category

```r
setup1[, c("pred.prob")]<- predict(m1, newdata=setup1, type="class")
setup1
   x1        x2 x3  pred.prob
1 0.6480006 0.1338694  1  Disag
2 0.6480006 0.1338694  2  Disag
```

These are the predicted categories given the new data
# Load package "erer", use function ocMe() for marginal effects

library(erer)

x <- ocME(m1, x.mean=TRUE)

x

<table>
<thead>
<tr>
<th></th>
<th>effect.Str</th>
<th>agree</th>
<th>effect.Agree</th>
<th>effect.Disag</th>
<th>effect.Str</th>
<th>disag</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>-0.198</td>
<td>-0.047</td>
<td>0.076</td>
<td>0.169</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>-0.050</td>
<td>-0.012</td>
<td>0.019</td>
<td>0.043</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>-0.018</td>
<td>-0.004</td>
<td>0.007</td>
<td>0.016</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# Type the following if you want t and p-values

x$out
Multinomial logit model

# Loading the required packages

library(foreign)
library(nnet)
library(stargazer)

# Getting the sample data from UCLA

mydata = read.dta("http://www.ats.ucla.edu/stat/data/hsb2.dta")

# Checking the output (dependent) variable

table(mydata$ses)

       low middle high
   47     95    58

# By default the first category is the reference.
# To change it so ‘middle’ is the reference type

mydata$ses2 = relevel(mydata$ses, ref = "middle")

NOTE: This section is based on the UCLA website [http://www.ats.ucla.edu/stat/r/dae/mlogit.htm](http://www.ats.ucla.edu/stat/r/dae/mlogit.htm), applied to data from the page [http://www.ats.ucla.edu/stat/stata/output/stata_mlogit_output.htm](http://www.ats.ucla.edu/stat/stata/output/stata_mlogit_output.htm). Results here reproduce the output in the latter to compare, and to provide an additional source to interpret outcomes.
# Running the multinomial logit model using the multinom() function

```r
multil = multinom(ses2 ~ science + socst + female, data=mydata)
```

```r
summary(multil)
```

Call:
multinom(formula = ses2 ~ science + socst + female, data = mydata)

Coefficients:
(Intercept) science socst femalefemale
low 1.912288 -0.02356494 -0.03892428 0.81659717
high -4.057284 0.02292179 0.04300323 -0.03287211

Std. Errors:
(Intercept) science socst femalefemale
low 1.127255 0.02097468 0.01951649 0.3909804
high 1.222937 0.02087182 0.01988933 0.3500151

Residual Deviance: 388.0697
AIC: 404.0697

These are the logit coefficients relative to the reference category. For example, under ‘science’, the -0.02 suggests that for one unit increase in ‘science’ score, the logit coefficient for ‘low’ relative to ‘middle’ will go down by that amount, -0.02.

In other words, if your science score increases one unit, your chances of staying in the middle ses category are higher compared to staying in low ses.
Multinomial logit model

# The multinom() function does not provide p-values, you can get significance of the coefficients using the stargazer() function from the package -stargazer.
# The model will be saved in the working directory under the name 'multi1.htm' which you can open with Word or any other word processor.

library(stargazer)
stargazer(multi1, type="html", out="multi1.htm")

NOTE: Use the option type = "text" if you want to see the results directly in the RStudio console.
Multinomial logit model: relative risk ratios

# Relative risk ratios allow an easier interpretation of the logit coefficients. They are the exponentiated value of the logit coefficients.

multil.rrr = exp(coef(multi1))

```
multi1.rrr
       (Intercept)   science    socst femalefemale
low   6.76855944 0.9767105  0.9618235    2.2627869
high  0.01729593 1.0231865 1.0439413    0.9676623
```

library(stargazer)
stargazer(multi1, type="html", coef=list(multi1.rrr), p.auto=FALSE, out="multi1rrr.htm")

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Keeping all other variables constant, if your science score increases one unit, you are 0.97 times more likely to stay in the low ses category as compared to the middle ses category (the risk or odds is 3% lower). The coefficient, however, is not significant.

Keeping all other variables constant, if your science score increases one unit, you are 1.02 times more likely to stay in the high ses category as compared to the middle ses category (the risk or odds is 2% higher). The coefficient, however, is not significant.

**NOTE:** Use the option `type = "text"` if you want to see the results directly in the RStudio console.
Ordinal logit model: predicted probabilities

# At specific values, example science and socst at their means for males and females.
# Use "probs" for predicted probabilities given specific predictors

```r
allmean <- data.frame(science=rep(mean(mydata$science),2),
                      socst=rep(mean(mydata$socst),2),
                      female = c("male","female"))

allmean

 science socst female
1  51.85 52.405   male
2  51.85 52.405 female

allmean[, c("pred.prob")]<-

allmean

 science socst female pred.prob.middle pred.prob.low pred.prob.high
1  51.85 52.405   male        0.5555769     0.1441171      0.3003061
2  51.85 52.405 female        0.4739293     0.2781816      0.2478890

# Use "class" for the predicted category

allmean[, c("pred.prob")]<-

allmean

 science socst female pred.prob
1  51.85 52.405   male  middle
2  51.85 52.405 female  middle
```

Setup for new predicted probabilities

These are the predicted categories given the new data
Sources


UCLA, [http://www.ats.ucla.edu/stat/r/dae/](http://www.ats.ucla.edu/stat/r/dae/)

StatsExchange, [http://stats.stackexchange.com/](http://stats.stackexchange.com/)

R packages:
- mfx- [http://cran.r-project.org/web/packages/mfx/mfx.pdf](http://cran.r-project.org/web/packages/mfx/mfx.pdf)
- erer- [http://cran.r-project.org/web/packages/erer/erer.pdf](http://cran.r-project.org/web/packages/erer/erer.pdf)